PROYECTO - De JavaScript a Angular

Debes pensar en un sistema donde haya dos entidades que tengan una relación 1:N. Como ejemplo:

- Cliente Pedidos
- Empresa Empleados
- Autor Libros

Objetivo del Proyecto:

- 1. Diseña un **CRUD** para cada una de las entidades.
- Crea una estructura modular y escalable que mantenga el proyecto organizado y
 facilite su mantenimiento y expansión, tal y como se hace en entornos de desarrollo
 como Angular.

Requisitos de la Interfaz y Funcionalidad:

Entidad Principal:

- **Formulario para Crear**: Un formulario para añadir nuevos elementos en la entidad principal.
- Listado: Una vista en la que se muestren todas las instancias de la entidad principal.
 - Dentro de este listado, deben aparecer opciones para editar y eliminar cada elemento.
 - Restricción: No se podrá eliminar un elemento de la entidad principal si hay datos en la entidad secundaria que dependan de él.

Entidad Secundaria:

- Formulario para Crear: Un formulario para añadir nuevos elementos en la entidad secundaria, incluyendo los controles y funcionalidad necesaria para especificar con qué entidad principal se relaciona.
- Listado:
 - Ver todas las instancias de la entidad secundaria.
 - Ver las instancias relacionadas con una entidad principal en particular.
 - o Las opciones de **editar** y **eliminar** estarán disponibles en el listado.

Implementación de las Clases:

Hasta ahora, hemos utilizado objetos literales para representar los datos en JavaScript, como en los siguientes ejemplos:

```
const person = {};
person.name = "John";

const objectName = {
  member1Name: member1Value,
  member2Name: member2Value,
  member3Name: member3Value
};
```

JavaScript object basics - Learn web development | MDN

Para este proyecto, **debes crear las clases correspondientes para cada entidad**, con las propiedades y métodos apropiados, como en el siguiente ejemplo:

```
class Rectangle {
  constructor(height, width) {
      this.height = height;
      this.width = width;
  }
}
```

Classes - JavaScript | MDN

Tipos de Datos a Incluir:

- 1. Cadenas de texto
- 2. Números
- 3. Coordenadas geográficas (latitud y longitud).
- 4. **Booleanos** (true/false)
- 5. Otros tipos de datos que se adapten al contexto de tus entidades.

Carga inicial de datos:

La **carga inicial de datos** se realizará a través de la lectura de dos archivos . j son (uno para cada entidad) que estarán ubicados en la carpeta assets.

Estructura del Proyecto:

Crea una estructura de directorios organizada y modular que facilite la escalabilidad y el mantenimiento del proyecto, cada entidad debe tener su propia carpeta con todos los archivos necesarios para su gestión, así se mantiene una clara separación de responsabilidades y se crea un proyecto claro y escalable.

Los datos de los archivos JSON se almacenan en una carpeta assets separada, ideal para facilitar la carga y el acceso a los datos.

Para un proyecto de ejemplo que podría ser "Empresa-Empleados", la estructura podría ser similar a esta:

proyecto-empresa-empleados/ - src/ - empresa/ — Empresa.js # Clase Empresa empresaController.js # Controlador para gestionar CRUD de empresas # Página HTML para listar empresas listaEmpresas.html crearEmpresa.html # Página HTML para crear una nueva empresa editarEmpresa.html # Página HTML para editar una empresa # CSS específico para las páginas de empresa - empresa.css empleado/ - Empleado.is # Clase Empleado - empleadoController.js # Controlador para gestionar CRUD de empleados listaEmpleados.html # Página HTML para listar empleados - crearEmpleado.html # Página HTML para crear un nuevo empleado - editarEmpleado.html # Página HTML para editar un empleado - empleado.css # CSS específico para las páginas de empleado # Página de inicio con enlaces a los CRUDs - index.html # JavaScript principal para la lógica de la interfaz - main.js - css/ └── main.css # CSS global para todo el proyecto - assets/ empresas.json # Archivo JSON para almacenar datos de empresas empleados.json # Archivo JSON para almacenar datos de empleados - package.json # Información y dependencias del proyecto - README.md # Documentación del proyecto

Instrucciones para entregar el proyecto

- 1. Crea un repositorio en GitHub para el proyecto
- 2. Indica el enlace al repositorio y cualquier otra información que sea necesaria para que acceda a él y que lo descargue.

Consejos:

- Mantén tu repositorio actualizado y organizado, haciendo commits frecuentemente para guardar tu progreso.
- Usa nombres descriptivos en los mensajes de tus commits para que tú y tu profesora puedan ver fácilmente los cambios realizados.
- Lee y actualiza el archivo README. md para que incluya instrucciones y detalles importantes sobre tu proyecto. Esto es útil para que cualquiera que vea tu repositorio pueda entender cómo funciona el proyecto.